

Files

🔍

📁

⋮

🔍 linkedList.java ⋮

🔍 Main.java

🔍 wordNode.java

☰ words.txt

🔍

📄

🔒

⚙️

📁

✓

linkedList.java ×

83

}

84

85

//Delete Word

86

void deleteWord(String word) {

87

wordNode curr,prev;

88

curr = front;

89

prev = front;

90

int wordCode = wordCodeFinder(word);

91

while(curr.next != null && curr.wordCode != wordCode){

92

prev = curr;

93

curr = curr.next;

94

}

95

if(curr.wordCode == wordCode){

96

prev.next = curr.next;

97

curr.next = null;

98

}

99

}//END DELETE

100

101

// Make Node

102

public wordNode makeNode(String word) {

103

wordNode n = new wordNode();

104

n.wordCode = wordCodeFinder(word);

105

n.word = word;

106

n.count = 1;

107

n.next = null;

108

return n;

109

}

110

111

// ShowList

112

void showList() {

113

wordNode curr = front;

114

while (curr != null) {

115

System.out.println(curr.word + " " + curr.wordCode);

116

curr = curr.next;

117

}

118

}

119

120

}

121

?

CPU

RAM ⚠️

Storage

⬆️

Files

linkedList.java

Main.java

wordNode.java

words.txt

Main.java x

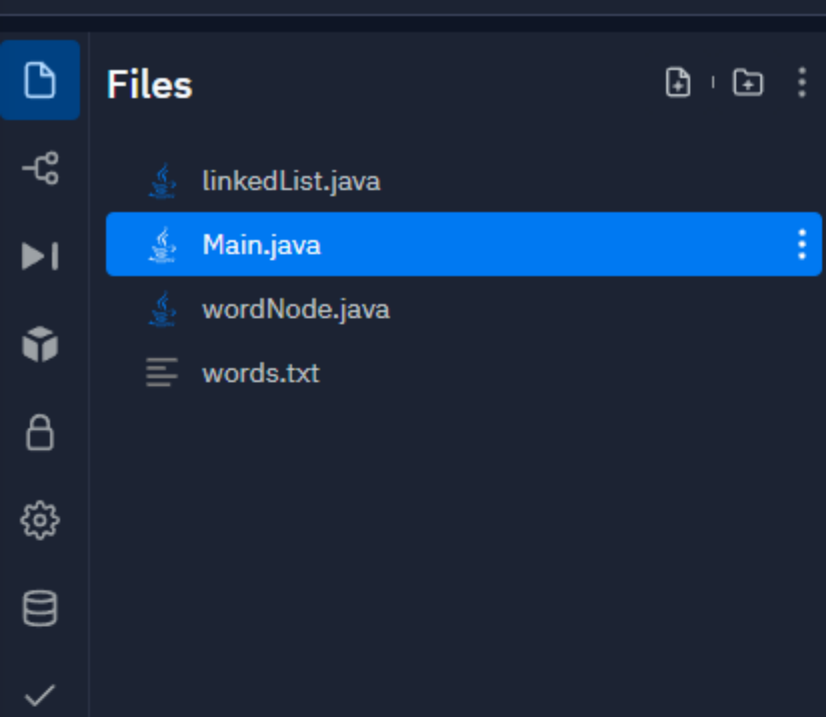
```
1 import java.util.Scanner;
2 import java.io.*;
3
4 class Main {
5     public static String fName;
6
7     public static void main(String[] args) {
8         linkedList l = new linkedList();
9         Scanner scan = new Scanner(System.in);
10        boolean repeating = true;
11        getFileName();
12        readFileContents(l);
13
14        while(repeating ==true){
15            System.out.println("***** MENU *****");
16            System.out.println("1. Print out dictionary.");
17            System.out.println("2. Add a word to dictionary.");
18            System.out.println("3. Delete a word from dictionary.");
19            System.out.println("4. List all words that start with a certain letter.");
20            System.out.println("5. Exit.");
21            System.out.println("*****");
22            //Build the linked list
23            int choice = scan.nextInt();
24
25            if(choice ==1){
26                //SHOWLIST FUNCTION
27                l.showList();
28            }
29
30            if(choice ==2){
31                System.out.println("What word would you like to be added?");
32                String wordToAdd = scan.next();
33                //ADD FUNCTION
34                l.insert(wordToAdd, 0);
35                l.showList();
36            }
37            if(choice ==3){
38                System.out.println("What word would you like to be deleted?");
39                String wordToDelete = scan.next();
40                //DELETE FUNCTION
41                l.deleteWord(wordToDelete);
42                l.showList();
43            }
44            if(choice ==4){
45                System.out.println("What is the letter of the words you would like to see?");
46                char findChapter = scan.next().charAt(0);
47                System.out.println("All the words starting with " +findChapter + ":");
48                l.chaptersByLetter(findChapter);
49            }
50            if(choice == 5){
51                System.out.println("Goodbye!");
52                repeating = false;
53            }
54        }
55        scan.close();
56    }
57
58    public static void getFileName(){
```

?

CPU

RAM

Storage



```

Main.java x
like to see?");
46 char findChapter = scan.next().charAt(0);
47 System.out.println("All the words starting with " + findChapter +
":");
48 l.chaptersByLetter(findChapter);
49 }
50 if(choice == 5){
51 System.out.println("Goodbye!");
52 repeating = false;
53 }
54 }
55 scan.close();
56 }
57
58 public static void getFileName(){
59 Scanner in = new Scanner(System.in);
60 System.out.println("Enter file name please.");
61 fName = in.nextLine();
62 System.out.println("You entered " + fName);
63 }
64 public static void readFileContents(linkedList l){
65
66 boolean looping;
67 String word;
68 DataInputStream in;
69 String line;
70 int i = 0;
71 String finalWord = "";
72 try{
73 Scanner scan = new Scanner(new File(fName));
74 while(scan.hasNextLine()){
75 l.insert(scan.nextLine(), i);
76 }
77 scan.close();
78
79 }
80 catch(IOException e){
81 System.out.println("Error " + e);
82 }
83
84 }
85 }
86

```

```

Console Shell
> sh -c javac -classpath .:target/dependency/* -d . $(find . -type f -name '*.java')
> java -classpath .:target/dependency/* Main
Enter file name please.
words.txt
You entered words.txt
***** MENU *****
1. Print out dictionary.
2. Add a word to dictionary.
3. Delete a word from dictionary.
4. List all words that start with a certain letter.
5. Exit.
*****
1
a 66404
an 68464
All 68488
also 68495
and 68532
because 68973
big 69081
bird. 69092
cats; 69562
chance. 69725
day 70243
deal 70323
depends. 70338
dictionary 70429
dogs 70589
egret. 71068
Every 71445
fishing. 71797
for 71952
friends 72021
fun. 72104
get 72370
go 72546
Halloween 72934
hang 72936
have 72944
horses 73304
in 73872
interesting 73956
is 74002
It 74028
just 74813
kinds 75172
like 75845
lives 75856
lots 76010
Love, 76012
lyrics. 76268
many 76316
married 76320
of 77720
on 77928
One 77997
other 78156
out 78194
prefer 78777
random 79696
ride 79894
She 80545
sing 80580
Sometimes 80735
songs 80736
strange 80870
that 81217

```

Files

linkedList.java

Main.java

wordNode.java

words.txt

CPU

RAM

Storage

```
Main.java x
    like to see?");
46     char findChapter = scan.next().charAt(0);
47     System.out.println("All the words starting with " +findChapter +
    ":");
48     l.chaptersByLetter(findChapter);
49     }
50     if(choice == 5){
51         System.out.println("Goodbye!");
52         repeating = false;
53     }
54     }
55     scan.close();
56     }
57
58     public static void getFileName(){
59         Scanner in = new Scanner(System.in);
60         System.out.println("Enter file name please.");
61         fName = in.nextLine();
62         System.out.println("You entered " + fName);
63     }
64     public static void readFileContents(linkedList l){
65
66         boolean looping;
67         String word;
68         DataInputStream in;
69         String line;
70         int i = 0;
71         String finalWord = "";
72         try{
73             Scanner scan = new Scanner(new File(fName));
74             while(scan.hasNextLine()){
75                 l.insert(scan.nextLine(),i);
76             }
77             scan.close();
78
79         }
80         catch(IOException e){
81             System.out.println("Error " + e);
82         }
83     }
84 }
85 }
86
```

ConsoleShell

zoo. 85469
***** MENU *****
1. Print out dictionary.
2. Add a word to dictionary.
3. Delete a word from dictionary.
4. List all words that start with a certain letter.
5. Exit.

3
What word would you like to be deleted?
day
a 66404
an 68464
All 68488
also 68495
and 68532
because 68973
big 69081
bird. 69092
cats; 69562
chance. 69725
deal 70323
depends. 70338
dictionary 70429
dogs 70589
egret. 71068
Every 71445
fishing. 71797
for 71952
friends 72021
fun. 72104
get 72370
go 72546
Halloween 72934
hang 72936
have 72944
horses 73304
in 73872
interesting 73956
is 74002
It 74028
just 74813
kinds 75172
like 75845
lives 75856
lots 76010
Love, 76012
lyrics. 76268
many 76316
married 76320
of 77720
on 77928
One 77997
other 78156
out 78194
prefer 78777
random 79696
ride 79894
She 80545
sing 80580
Sometimes 80735
songs 80736
strange 80870
that 81217
the 81221
this 81225
thoughts 81231

Files

linkedList.java

Main.java

wordNode.java

words.txt

CPU

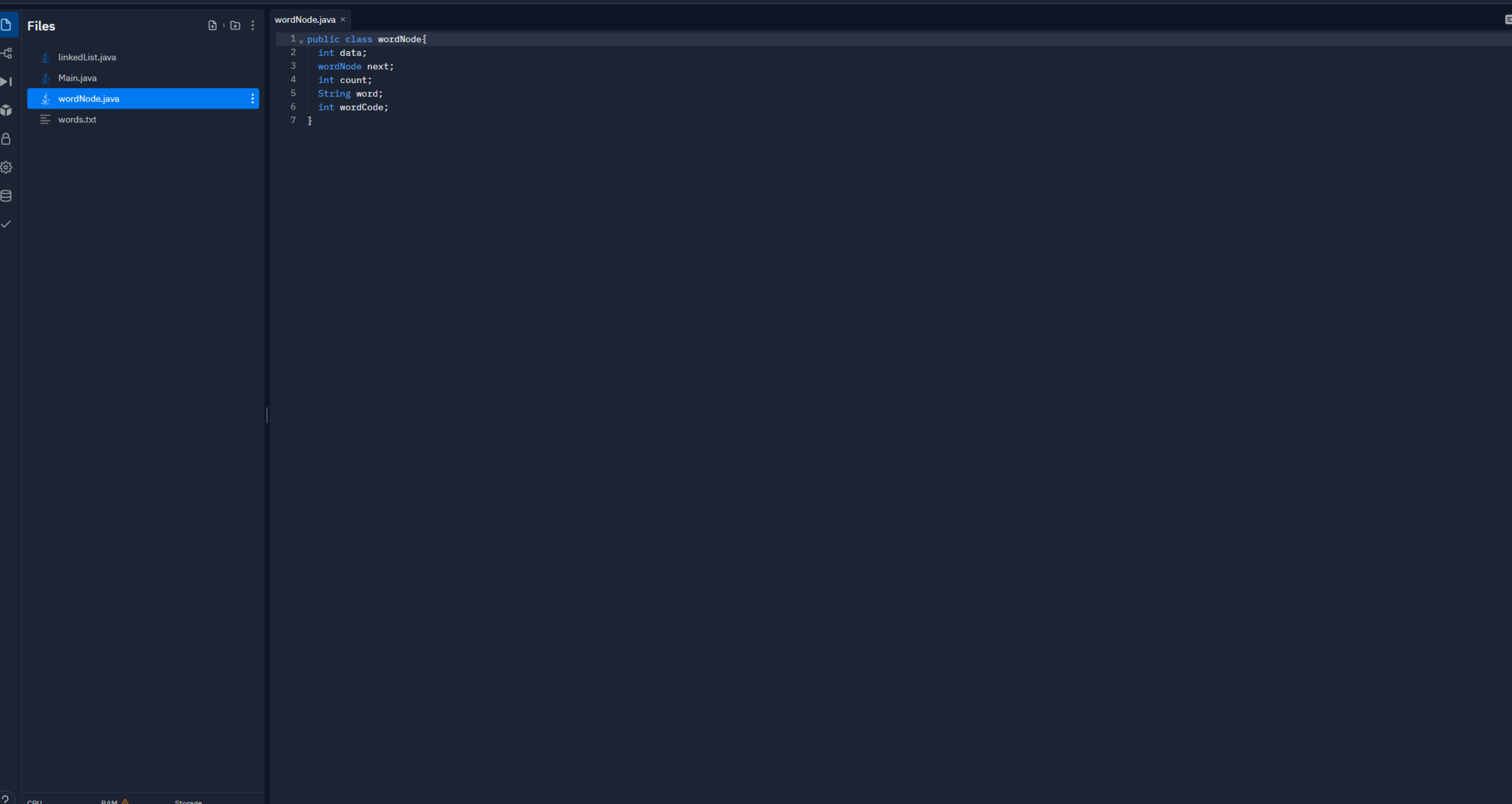
RAM

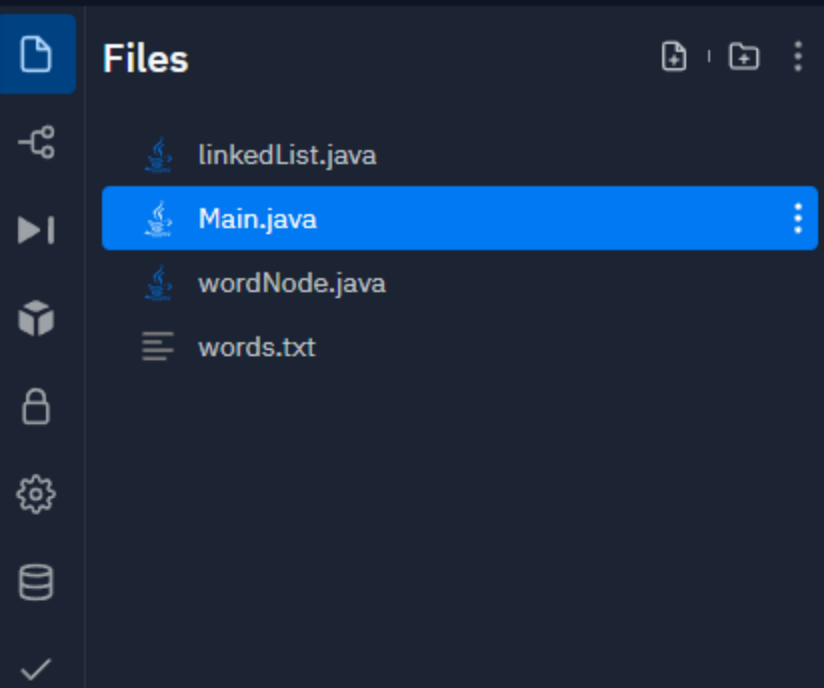
Storage

```
Main.java x
like to see?");
46     char findChapter = scan.next().charAt(0);
47     System.out.println("All the words starting with " +findChapter +
    ":"");
48     l.chaptersByLetter(findChapter);
49 }
50 v if(choice == 5){
51     System.out.println("Goodbye!");
52     repeating = false;
53 }
54 }
55 scan.close();
56 }
57
58 v public static void getFileName(){
59     Scanner in = new Scanner(System.in);
60     System.out.println("Enter file name please.");
61     fName = in.nextLine();
62     System.out.println("You entered "+ fName);
63 }
64 v public static void readFileContents(linkedList l){
65
66     boolean looping;
67     String word;
68     DataInputStream in;
69     String line;
70     int i = 0;
71     String finalWord = "";
72 v try{
73     Scanner scan = new Scanner(new File(fName));
74 v while(scan.hasNextLine()){
75     l.insert(scan.nextLine(),i);
76 }
77 scan.close();
78
79 }
80 v catch(IOException e){
81     System.out.println("Error " + e);
82 }
83
84 }
85 }
86
```

ConsoleShell

```
> sh -c javac -classpath ./target/dependency/* -d . $(find . -type f -name '*.java')
> java -classpath ./target/dependency/* Main
Enter file name please.
words.txt
You entered words.txt
***** MENU *****
1. Print out dictionary.
2. Add a word to dictionary.
3. Delete a word from dictionary.
4. List all words that start with a certain letter.
5. Exit.
*****
4
What is the letter of the words you would like to see?
t
All the words starting with t:
that 81217
the 81221
this 81225
thoughts 81231
to 81334
together. 81405
***** MENU *****
1. Print out dictionary.
2. Add a word to dictionary.
3. Delete a word from dictionary.
4. List all words that start with a certain letter.
5. Exit.
*****
5
Goodbye!
> []
```



Main.java x

```

48         l.chaptersByLetter(findChapter);
49     }
50     if(choice == 5){
51         System.out.println("Goodbye!");
52         repeating = false;
53     }
54 }
55 scan.close();
56 }
57
58 public static void getFileName(){
59     Scanner in = new Scanner(System.in);
60     System.out.println("Enter file name please.");
61     fName = in.nextLine();
62     System.out.println("You entered " + fName);
63 }
64 public static void readFileContents(linkedList l){
65
66     boolean looping;
67     String word;
68     DataInputStream in;
69     String line;
70     int i = 0;
71     String finalWord = "";
72     try{
73         Scanner scan = new Scanner(new File(fName));
74         while(scan.hasNextLine()){
75             l.insert(scan.nextLine(), i);
76         }
77         scan.close();
78
79     }
80     catch(IOException e){
81         System.out.println("Error " + e);
82     }
83
84 }
85 }
86

```


Files

linkedList.java

Main.java

wordNode.java

words.txt

CPU

RAM

Storage

LinkedList.java ×

```
58     searching = true;
59     while (searching) {
60         if (curr.wordCode == wordCode) {
61             searching = false;
62         } else {
63             if (curr.wordCode < wordCode) {
64                 if(curr.next == null){
65                     temp = makeNode(word);
66                     curr.next = temp;
67                     searching = false;
68                 } else {
69                     prev = curr;
70                     curr = curr.next;
71                 }
72             } // End else if
73         }
74         if (curr.wordCode > wordCode) {
75             temp = makeNode(word);
76             temp.next = curr;
77             prev.next = temp;
78             searching = false;
79         }
80     }
81     } // End while
82 } // End else
83 }
84
85 //Delete Word
86 void deleteWord(String word) {
87     wordNode curr,prev;
88     curr = front;
89     prev = front;
90     int wordCode = wordCodeFinder(word);
91     while(curr.next != null && curr.wordCode != wordCode){
92         prev = curr;
93         curr = curr.next;
94     }
95     if(curr.wordCode == wordCode){
96         prev.next = curr.next;
97         curr.next = null;
98     }
99 }//END DELETE
100
101 // Make Node
102 public wordNode makeNode(String word) {
103     wordNode n = new wordNode();
104     n.wordCode = wordCodeFinder(word);
105     n.word = word;
106     n.count = 1;
107     n.next = null;
108     return n;
109 }
110
111 // ShowList
112 void showList() {
113     wordNode curr = front;
114     while (curr != null) {
115         System.out.println(curr.word + " " + curr.wordCode);
```